
TIMEX SINCLAIR USERS GROUP MILE HIGH CHAPTER

C/O FRANK HOLLAND, 1423 S. PEARL ST., DENVER CO 80210, 733-8103

NOV/DEC 1987 *HAPPY HOLIDAYS TO ALL AND A PROSPEROUS NEW YEAR*

TSUG meetings are USUALLY on the 4th Thursday of every month.
The DEC. meeting is cancelled because of the holidays.

MILE HI TSUG maintains a sub-board on THE KING'S MARKET BBS.
1-303-665-6091, 8-1-NONE. Accessible thru PC-PURSUIT.

MENU SELECTIONS TO GET TO THE SUB-BOARD ARE:

- (1) CONTENTS
- (2) INTERESTS & USER GROUPS
- (3) TIMEX-SINCLAIR

PROPERTY OF
CCAT/S
655-7404

General messages for club members are addressed to "ALL".

MISC NOTES: MONITOR PROBLEMS

For those of you who have been having problems with monitors.
The "tearing" problem at the upper left corner of the display
seems to be because of a peculiar mis-match that can occur
between the computer and the monitor.

The 75 ohm impedance matching resistor and the 100mfd DC
isolation capacitor in the output circuit of the computer AND
the 75 ohm impedance matching resistor in the input of the
monitor form a "pi" network which feeds the input capacitor of
the monitor. This input capacitor seems to vary widely from
brand to brand and changes the balance of the "pi" network.

My solution was to remove the 100 mfd output capacitor (C-58)
and substitute a 470 mfd capacitor (after a lot of
experimentation). You may need to experiment a bit for the
particular monitor that you have; mine is an AMDEK 300-A.

There wasn't enough room in the video cage for the new cap so
I replaced the capacitor with a jumper and made an add-on plug
box with the 470 mfd cap. This goes between the "monitor out"
plug and the monitor.

I have not experimented with replacing the 75 ohm resistors
with 300 ohms for a different impedance match) because I would
need to tear into both the computer and the monitor.

If your monitor has a 75/300 ohm switch, then it would be a
simple matter to remove the entire output capacitor/resistor
combination from the 2068, then add a switch and the two (75/300
ohm) resistors to the capacitor in the add-on box so that you
have an impedance selection switch for the computer.

Just remember that the emitter resistor and blocking
capacitor are now in the external box. If you power-up the 2068
(or connect a monitor) without them it will damage the computer.

And always make sure that the two switches are set the same.

OS-64 BUG: From SYNCWARE NEWS, Sept-Oct '87

OS-64 cartridge version 1.72 sets RAMTOP at 65255. This is ABOVE the machine stack and ram-resident code. If a long program over-writes the machine stack, the system will crash.

The correct place for RAMTOP in double display mode is 63255, the first byte below UDG.

"CLEAR 63255" will reset RAMTOP and allow use of the UDG's.

"CLEAR 63423" will set it above the UDG area and give you 168 more bytes to use.

I found the same thing in my version 1.7 cartridge. Also, the default interface settings are different to the later cartridges (see the following documentation "LINE CHANGES AND ADDITIONS", lines 9994 and 9997.

THE FEATURE ARTICLE

or

Why you didn't receive a newsletter last month

Several months ago I mentioned that I had rewritten PRO/FILE so that it could be used with the OS-64 cartridge in the 2068. Well, I think that I have finally ironed out all of the bugs and here it is.

The explanatory notes and the BASIC program are in this newsletter. The BASIC program is also in the DOWNLOAD section of the King's Market BBS.

The MC portion of the program has not been included because it is a copyrighted program and you have to pay Tom Woods for it.

I have included instructions on how to change it if you already have a legal copy.

I have sent a copy of the entire program to Tom, so write to him if you don't have PRO/FILE and are interested in it.

WARNING:

My printer makes lower case "l"s and the number "1" look very similar. Also the letter "o" and the number "0".

Both are used in the BASIC listing and it is very easy to confuse them. BE CAREFUL.

TILL NEXT MONTH

Frank

THESE NOTES ARE INTENDED AS A SUPPLEMENT TO THE PROFILE MANUAL.

I have rewritten PRO/FILE 2068 to operate with the ZEBRA OS-64 module to provide a 64 column by 15 line data-file and display. Portions of the BASIC program were changed and the machine code had to be moved (and changed) to accomodate the changes in the operating system.

There are only about 500 bytes left unused in ram.

FILE SIZE:

I have used 22760 bytes for d\$ (maximum file size) because this rewrite was also intended for use with the A&J Microdrive (which makes a smaller file size practical) and, at this size, I can get 4 complete files on a 62 foot wafer.

I like to SAVE a standard size file so that changes will not cause a new SAVE to over-write part of the following file on a wafer.

ADDITIONS:

I have added the following routines from the PRO/FILE manual (pps 97-107);

- The keypress "tick"
- keypress non-auto-repeat
- lprint from edit (copy display)
- autosearch improvement
- auto-sort (by 32 chars)
- block purge.

I have also added a routine to allow 2 files to be merged.

CHANGES:

The main menu has been completely rearranged and several new functions added.

The data-entry/edit menu has been rearranged and all of the functions now appear on a single menu. Upper/lower case and insert/over conditions now appear on the menu. The line and column positions of the cursor are also displayed on the menu along with the LPRINT format.

"STOP" to exit the data-entry mode and "STEP" to exit the edit mode are displayed as a reminder. The program is supposed to differentiate between them so BUT BE CAREFUL. I have experienced random problems using STOP from edit.

Different key-click sounds are provided for data-entry and control functions.

LINE CHANGES AND ADDITIONS:

Many of the new line numbers and routines will not match the original program but they are still obvious enough that you can sort them out. These are the major changes.

Lines 20 - 118: main menu & select
Lines 1050 - 2000: data review menu
Lines 5004 - 5018: data enter/edit menu
Lines 5500 - 5510: data "LOAD"
Lines 5600 - 5630: data "SAVE"
Lines 5650 - 5680: "MERGE" two files
Lines 7200 - 7240: "LPRINT"
Line 9800: "CHANGE FILENAME"
Line 9994: CLEAR 62627 and LOAD mc.
Line 9995: DIM (22760) and certain standard values are put into variables to save ram space.
Line 9996: Housekeeping variables are initialized including the blank lines and "zzz"s that are necessary for sorting and printing operations. GOSUB 5620 is to insert the "*SEARCH IS COMPLETE*" message in the first 20 bytes of d\$ (this is also used by LOAD.
Line 9997: POKE 24201,0 (A&J drive 0)
POKE 65510,0 (turn off line feed, OS-64)
POKE 65523,185 (select A&J interface, OS-64)
POKE 65525,72 (LPRINT # of chars per line,
plus # of chars in margin, OS-64)
RANDOMIZE USR 15623 (initialize # of chars, OS-64)
Line 9998: initialize y\$ & f\$, start (GO TO 1)
Line 9999: SAVE & VERIFY for backup

You may need to make some changes in Line 9997 to match your OS-64 (they'r not all the same), your interface, and your printer.

MACHINE CODE:

The PRO FILE MC now starts at 62628 (down 861 bytes) because the operating system is moved around for the 64 column mode and portions of it now occupy the top of ram.

You will need HOT-Z, or some other assembler/disassembler that will allow you to relocate blocks of code. Otherwise you will have to go thru the MC and change all of the GOTOs & JPs by hand.

It was only necessary to change 3 bytes to accomodate the 64 column mode. Addresses are the NEW locations.

F502 = 40 (40 hex = 64 spaces)
F463 = 40 (set count to 64)
F4C4 = 41 (# of columns + 1)

"SAVE" and "LOAD":

Both SAVE and LOAD will automatically put the computer in CAPS mode (POKE 23658,8).

All data SAVE/LOAD operations are set up for A&J drive 0 because I always keep programs in drive 1 and files in drive 0.

The dimensioned data file (d\$), as the first variable declared, is always located at (PEEK 23627+256*PEEK 23628). The first 6 bytes are header information, therefore the actual data starts at byte 7. Line 5630 creates variable "loc" which is the beginning of the data. loc=6+(PEEK 23627+256*PEEK 23628).

"SAVE"; Line 5600; will use the "FILENAME" (f\$) that appears on the main menu. The routine will also ask for a file number.

The message "SEARCH IS COMPLETE" will be removed from d\$ and the value of variable "p" (total bytes used) will be inserted in the, now unused, 20 bytes of the header.

The program will automatically VERIFY.

"LOAD"; Line 5500; will ask "HAS CURRENT FILE BEEN SAVED?" (as a safety check to prevent losing a file). "N" will return you to the main menu, "Y" will ask for the filename to be loaded.

Variable "p" is extracted from the new d\$ and the message "*SEARCH IS COMPLETE*" is replaced in the first 20 bytes.

"LPRINT":

The OS-64 operating system does not support the 2040 printer. However, it does provide a centronics driver for a large printer (interface selectable). Therefore, the driver code used in the manual is not necessary.

The total number of characters per line are set in Line 9997. However, since some printers do not have a "left margin" control - Line 7220 contains the necessary number of blank spaces to create a left margin. This means that line 9997 (POKE 65525,72) is the number of characters to be printed PLUS the number of characters for the margin.

If you want a blank space between each individual file printout, then change line 7225 to "NEXT x: LPRINT".

EXAMPLE; I have indexed the club newsletter library. The first line is always the newsletter name and date with content notes on the other lines. I get rid of the LPRINT when I want a list of just the newsletter names & dates. I use the LPRINT when I want a printout of the entire file, with a space between each file.

COMBINING FILES (MERGE):

PRO/FILE allows deleting blocks of files so that large files may be split. I have added a routine that allows two files to be combined. As written, it requires 2 microdrives.

EDIT THE TWO FILES TO THE SIZE AND DATA THAT YOU WANT.

Remembered that the total number of bytes in the two files MUST BE LESS than the maximum file size (22760 bytes minus 20, or 11375 bytes each).

Place the wafer containing the regular files in drive 0, place a formatted wafer (20-35 ft) in drive 1.

When you select "MERGE", the program will prompt for the first filename. It will then LOAD that file (from drive 0), place the actual file size in variable "p1" ($p1=p-20$), place the filename in "h\$", and then SAVE (and VERIFY) the file (at its actual file size minus the 20 byte header) to drive 1. See line 5660, SAVE "a1,"+ f\$ CODE (loc+20),p1.

The program will then prompt for the next filename (to be LOADED from drive 0). This will LOAD normally.

The first file will then automatically be reLOADED (using the name in "h\$), AND using the new value in variable "p" to determine where the file should go in d\$ so that it is immediately behind the resident file. Line 5680, LOAD "a"+h\$ CODE (loc+p+1)

The value in variable "p1" is then added to the value in variable "p" to give the size of the combined file.

It is a simple matter to rewrite the program if you only have one microdrive. You will need to put pauses (and reminder messages) in to allow time to swap wafers.

WARNING: It has been my experience that no two microdrives run at the same speed. Therefore, if you have saved several files using the slower drive and then over-write one of those files using the faster drive, it will also over-write the beginning of the following file.

```

0001  REM  ** PRO/64 **  MODIFIED OCT, 1987 BY FRANK HOLLAND

0003  LET a=t0:
      LET x$=""

0004  RANDOMIZE p:
      POKE VAL "63174", PEEK VAL "23670":
      POKE VAL "63175", PEEK VAL "23671":
      RANDOMIZE USR VAL "63407"

0005  LET y=t0:
      DIM e$(t15, t64):
      CLS

0008  IF PEEK VAL "62750" = t0
      AND y$ <> "D"
      AND y$ <> " STEP "
      THEN GO SUB VAL "7300"

0010  IF y$ = "D"
      OR y$ = " STEP "
      THEN GO TO t119

0020  CLS:
      PRINT AT t1, t15; "***** PRO/FILE 2068 + OS-64 *****"

0040  PRINT AT t3, t3; "FILE : "; f$
      "FREE : "; LEN d$ - p; "bytes"
      TAB t3; "ORDER : "; s
      "FORMAT: "; a$

0050  PRINT AT VAL "7", t10; "<*> LIST ALL FILES"
      TAB t10; "<E>nter NEW FILE"
      TAB t10; "<S>ave FILES (drive 0)"
      TAB t10; "<L>oad FILES (drive 0)"
      TAB t10; "<M>erge FILES"
      TAB t10; "<A>utosearch (& lprint)"
      TAB t10; "<F>ormat LPRINT"
      TAB t10; "<C>hange FILENAME (7 chars.)"
      TAB t10; "<P>urge ALL FILES"

0051  PRINT AT VAL "18", t2; "SEARCH COMMAND\# (line 1-9) to sort files."
      TAB t2; "SEARCH COMMAND\D to delete series of files."
      TAB t2; "Separate multi-word command words with token "AND"."

0055  PLOT t0, t0:
      DRAW t255, t0:
      DRAW t0, VAL "175":
      DRAW -t255, t0:
      DRAW t0, VAL "-175"

0100  POKE VAL "23658", t8:
      INPUT "SEARCH COMMAND? "; x$

0101  IF x$ = "E"
      THEN GO TO VAL "5000"

```

```

2  IF x$ = "S"
    THEN GO SUB VAL "5600":
    GO TO t1

03 IF x$ = "L"
    THEN GO SUB VAL "5500":
    GO TO t1

04 IF x$ = "A"
    THEN GO TO VAL "5200"

05 IF x$ = "F"
    THEN GO SUB VAL "6500":
    GO TO t20

06 IF x$ = "C"
    THEN GO TO VAL "9800"

07 IF x$ = "P"
    THEN GO TO VAL "9650"

08 IF x$ = "M"
    THEN GO SUB VAL "5650":
    GO TO t1

15 IF x$ = ""
    THEN GO TO t1

16 IF x$(LEN x$) = " AND "
    OR LEN x$ > t64
    THEN GO TO t10

18 CLS

19 LET x$ = x$:
    LET x$ = x$

20 RANDOMIZE USR VAL "63179"

30 LET j$ = CHR$ PEEK VAL "23670"

40 IF a = t0
    THEN IF CODE j$ > VAL "48"
        AND CODE j$ < VAL "58"
        THEN LET s = VAL j$:
        LET a = t1:
        LET y = t0:
        LET y$ = " STEP ":
        GO SUB VAL "7300":
        LET p$ = b$:
        LET q$ = g$

50 LET b = USR VAL "63240"

```



```

0200 IF j$ = "D"
      THEN LET a = t1:
           LET s = t0:
           LET y$ = "D":
           LET y = t0

1000 PRINT AT t0,t0:
      DIM e$(t15, t64):
      LET e$(t1) = "":
      IF USR VAL "62714" THEN

1010 IF e$(t1, TO t18) = d$(t2 TO t19)
      AND b <> p
      THEN CLS:
           GO TO VAL "150"

1045 IF a
      THEN GO TO VAL "7000"

1047 IF a
      THEN GO SUB VAL "7300":
           GO SUB VAL "7310":
           GO SUB VAL "7300"

1049 IF e$(t1, TO t18) <> d$(t2 TO t19)
      AND CODE j$ > t32
      THEN GO TO VAL "8500" + CODE j$

1050 GO SUB VAL "9830":
      PRINT AT t17,t0: "<M>ain menu";
      TAB t25: "<E>dit file";
      TAB t45: "<P>rint file"
      "<ENTER> to continue";
      TAB t25: "<A>dd new file";
      TAB t45: "<F>ormat"
      "<R>estart same search";
      TAB t25: "<D>elete file";
      TAB t45: " "; a$

1055 PRINT AT t20, t1: "Search is for:";
      TAB VAL "36": "or enter new SEARCH COMMAND" x$
      POKE VAL "23658", t8

1056 INPUT "OPTION? "; y$

1060 IF y$ = ""
      AND b < p
      THEN CLS:
           GO TO VAL "150"

1070 IF y$ = "M"
      THEN GO TO t1

1075 IF USR VAL "63407" THEN
      (this line is supposed to be this way, it is not an error)

```

```

1080 IF PEEK VAL "63165" + t256 * PEEK VAL "63166"
    <> PEEK VAL "23627" + t256 * PEEK VAL "23628" + VAL "6"
    AND (y$ = "D" OR y$ = "E")
    THEN GO TO VAL "4000"

1090 IF y$ = "R"
    THEN CLS:
        LET x$ = x$
        GO TO t119 + (a * VAL "5121")

1100 IF y$ = "P"
    THEN GO TO VAL "7205"

1200 IF y$ = "F"
    THEN GO SUB VAL "9830":
        GO SUB VAL "6500":
        LET y$ = "":
        GO TO VAL "1050"

1400 IF a$ = "A"
    THEN LET x$ = "1":
        GO TO VAL "5000"

1500 LET x$ = y$:
    GO TO VAL "115"

2000 GO TO VAL "1050"

4000 LET p = p - USR VAL "62779":
    IF y$ = "D"
        THEN LET y$ = CHR$ (CODE "D" * a):
            GO TO VAL "4"

4010 GO TO VAL "5002"

5000 CLS:
    DIM e$(t15, t64)

5002 LET m = t0:
    LET z = VAL "23658":
    LET i = t0:
    LET c = t0

5003 GO SUB VAL "9830"

5004 PRINT AT t17,t0; "ARROWS move cursor",b$
    "SHIFT 0 = Delete character",
    "SHIFT 1 = Alternate Insert/Over"

5005 PRINT AT t19,t0; "SHIFT 2 = CAPS LOCK",
    "SHIFT 3 = LINE DELETE",
    "SHIFT 4 = LINE ERASE",
    "SHIFT SYMBOL SHIFT: Line insert",
    "<STOP> to close file":
    IF y$ = "E"
        OR y$ = "A"
        THEN PRINT AT VAL "21", T0; "<STEP> to return from ADD/EDIT"

```

```

5006 PRINT AT t16, t5; "UPPER CASE":
      IF PEEK z = t0
        THEN PRINT AT t16, t5; "lower case"

5007 PRINT AT t16, 50; "INSERT OFF":
      IF i = t1
        THEN PRINT AT t16, VAL 50; "INSERT ON "

5008 PRINT INVERSE t1; AT 1, c; SCREEN$ (1,c); INVERSE t0;
      AT t17, t32; "L=      ";
      TAB 35; 1 + 1;
      TAB 40; "C=      ";
      TAB 43; c+t1

5009 ON ERR GO TO 5008

5010 IF INKEY$ = ""
      THEN GO TO 5010

5015 LET y$ = INKEY$

5016 IF y$ = " STEP "
      OR y$ = " STOP "
      THEN ON ERR RESET:
        CLS:
        GO TO 6000

5017 IF y$ = ""
      THEN GO TO 5010

5018 BEEP .001, 50

5020 PRINT AT 1, c; SCREEN$ (1, c):
      IF CODE y$ < t16
        AND CODE y$ > t3
        THEN GO SUB 5100 + CODE y$:
          BEEP .005, 25:
          GO TO 5006

5025 IF i
      THEN LET e$(1+t1) = e$(1+t1 TO c) + " " + e$(1+t1, c+t1 TO 63):
        PRINT AT 1, t0; e$(1+t1)

5030 PRINT AT 1,c; y$:
      LET e$(1+t1, c+t1) = y$:
      LET c = (c+t1) * (c<63):
      LET 1 = 1 + (c=t0):
      LET 1 = 1 * (1<t15)

5050 GO TO 5008

```

```

5104  FOR x = 1 TO 13:
        LET e$(x+t1) = e$(x+t2):
        PRINT AT x, t0; e$(x+t1):
    NEXT x:
    LET e$(t15) = "":
    PRINT e$(t15):
    RETURN

5105  LET e$(1+t1),c+t1 TO) = "":
    PRINT AT 1,t0; e$(1+t1):
    RETURN

5106  POKE z, (PEEK z=t0) * t8:
    PRINT AT 1,c; e$(1+t1,c+t1):
    RETURN

5107  LET i = NOT i:
    PAUSE t5:
    RETURN

5108  LET c = c-(c>t0):
    RETURN

5109  LET c = c+(c< VAL "63"):
    RETURN

5110  LET 1 = 1+(VAL "14"):
    RETURN

5111  LET 1 = 1-(1>t0):
    RETURN

5112  LET e$(1+t1) = e$(1+t1, TO c) + e$(1+t1, c+t2 TO t63) + " ":
    PRINT AT 1, t0; e$(1+t1):
    GO TO VAL "5108"

5113  LET c = t0:
    LET 1 = 1 + t1:
    LET 1 = 1 - (1>VAL "14"):
    RETURN

5114  FOR x = t15 TO 1 + t2 STEP -t1:
        LET e$(x) = e$(x-t1):
        PRINT AT x - t1, t0; e$(x):
    NEXT x:
    LET e$(1+t1) = "":
    PRINT AT 1, t0; e$(1+t1):
    RETURN

5117  GO TO VAL "5004"

5130  RETURN

5200  GO SUB VAL "7300":
    LET a = t1:
    INPUT "PRINT OUT Y/N) "; y$:
    LET y = y$ = "Y"

```

```

5210 INPUT "SEARCH COMMAND? "; x$:
    IF x$ = ""
        OR LEN x$ > VAL "63"
        THEN BEEP VAL ".5", t10:
            GO TO VAL "5210"

5220 LET z$ = 0":
    INPUT "LINE NUMBER (0-15) "; z$:
    IF z$ = ""
        THEN BEEP VAL ".5", t10:
            GO TO VAL "5220"

5222 ON ERR GO TO VAL "5220":
    IF VAL z$ < t0
        OR VAL z$ > t15
        THEN GO TO VAL "5220"

5230 ON ERR RESET:
    LET s = VAL z$:
    CLS

5340 IF s <> t0
    THEN LET p$ = b$:
        LET q$ = g$

5350 GO TO t119

5500 CLS:
    POKE VAL "23658", t8:
    INPUT "HAS CURRENT FILE BEEN SAVED? (Y/N) "; y$:
    IF y$ <> "Y"
        THEN GO TO t1

5510 INPUT "FILE NAME? "; f$:
    GO SUB VAL "5630":
    LOAD "a" + f$ CODE loc:
    LET p = VAL d$ (t1 TO t20)

5111 GO TO VAL "5620"

5600 POKE VAL "23658", t8:
    CLS:
    INPUT "SAVE AT FILE #? "; n$:
    GO SUB VAL "5630":
    LET d$(t1 TO t20) = STR$ p

5610 SAVE "a" + n$ + "," + f$ CODE loc, VAL "21760":
    VERIFY "a" + f$ CODE

5620 LET d$ (t1 TO t20) = "*SEARCH IS COMPLETE*":
    RETURN

5630 LET loc = (PEEK VAL "23627" + t256 * PEEK VAL "23628") + VAL "6":
    RETURN

```

```

5650 CLS:
    PRINT "MERGE: file #1":
    GO SUB VAL "5510":
    LET p1 = p - t20:
    LET h$ = f$

5660 POKE VAL "24201", t1:
    SAVE "a1," + f$ CODE (loc + t20), p1:
    VERIFY "a" + f$ CODE:
    POKE VAL "24201", t0

5670 PRINT "MERGE: file #2":
    GO SUB VAL "5510"

5680 POKE VAL "24201", t1:
    LOAD "a" + h$ CODE (loc + p + t1):
    POKE VAL "24201", t0:
    LET p = p + p1:
    RETURN

6000 LET d$(p) = CHR$ (42 - a * 42):

6010 FOR x = t1 TO t15
    LET e$(x) = e$(x)
    IF USR 62628 THEN
    NEXT x

6020 LET p = USR 62669

6030 GO TO VAL "4" - (y$ = " STOP ") + (a * (y$ = " STEP "))

6500 PRINT AT t18, t2; "ENTER DESIRED FORMAT:"; b$
    TAB t2; "Type ALL or line numbers (1-9) separated by ""/""";
    TAB t2; "Use ""0"" to print blank lines"; b$

6510 LET c$ = "":
    LET n = t0:
    INPUT "FORMAT? "; a$:
    LET a$ = a$ + "/":
    IF a$ (t1) = "/"
    THEN GO TO VAL "6510"

6515 IF a$ = "ALL/"
    THEN FOR x = t1 TO t15
        LET c$ = CHR$ x:
    NEXT x:
    LET d = t3:
    RETURN

6520 LET d = t1:
    FOR x = t1 TO LEN a$
        IF a$(x) < "/"
        OR a$(x) > "9"
        THEN GO TO VAL "6510"

```



```

6525 IF x < LEN a$
      THEN IF a$(x TO x+t1) = "/"
            THEN GO TO VAL "6510"

6530 IF a$(x) = "/"
      THEN LET n = VAL a$(d TO x-t1):
            LET d = x + t1:
            IF n < t16
            THEN LET c$ = c$ + CHR$ n

6535 IF n > t15
      THEN GO TO VAL "6510"

6540 NEXT x

6600 RETURN

7000 IF s = t0
      THEN GO TO VAL "7202"

7005 IF b = p
      THEN GO TO VAL "7070"

7010 LET s$ = e$(s, TO t32)

7020 IF s$ = p$
      THEN GO TO VAL "7200"

7030 IF s$ > p$
      THEN IF s$ < q$
            THEN LET q$ = s$

7050 GO TO VAL "7230"

7070 IF p$ = g$
      THEN GO TO VAL "1047"

7080 LET p$ = q$:
      LET q$ = g$:
      CLS

7090 LET x$ = x$:
      GO TO t119

7200 POKE PEEK VAL "63165" + t256 * PEEK VAL "63166", t0

7202 IF y= t0
      THEN GO TO VAL "1047"

```

```

7205  FOR x = t1 TO LEN c$
7210      IF CODE c$(x) = t0
          THEN LPRINT
7220      IF CODE c$(x)
          AND e$(x) > b$ + b$
          THEN LPRINT "          " + "e$(CODEc$(x))
7225  NEXT x
7230  IF a
      AND b < p
      AND y$ <> "C"
      THEN CLS:
          GO TO VAL "150"
7240  GO TO VAL "1047"
7300  LET v = PEEK VAL "62750":
      POKE VAL "62750", VAL "215" - v:
      POKE VAL "62736", VAL "215" - v:
      RETURN
7310  PRINT AT t0, t0:
      LET e$(t1) = "":
      IF USR VAL "62714" THEN
          (this line is supposed to be this way, it is not an error)
7311  RETURN
8548  GO TO t1
8557  LET p = p - USR VAL "62779":
      GO TO VAL "6000"
8567  GO TO t1
8568  GO TO VAL "1060"
9650  INPUT "ARE YOU SURE? (Y/N) "; x$
      IF x$ = "y"
          OR x$ = "Y"
          THEN GO TO VAL "9995"
9660  GO TO t1
9800  INPUT "NEW FILENAME? "; f$:
      GO TO t1
9830      PLOT t0, VAL "54":
      DRAW t255, t0:
      PLOT t0, VAL "52":
      DRAW t255, t0:
      PRINT AT t16, VAL "22": "*** OPTION MENU ***":
      RETURN

```

```

9995 CLEAR:
DIM d$(VAL "22760");
LET t0 = VAL "0";
LET t1 = VAL "1";
LET t2 = VAL "2";
LET t3 = VAL "3";
LET t5 = VAL "5";
LET t8 = VAL "8";
LET t10 = VAL "10";
LET t15 = VAL "15";
LET t16 = VAL "16";
LET t17 = VAL "17";
LET t19 = VAL "19";
LET t20 = VAL "20";
LET t25 = VAL "25";
LET t32 = VAL "32";
LET t45 = VAL "45";
LET t64 = VAL "64";
LET t119 = VAL "119";
LET t255 = VAL "255";
LET t256 = VAL "256";

```

```
9997 POKE VAL "24201", t0:
      POKE VAL "65510", t0:
      POKE VAL "65523", VAL "185":
      POKE VAL "65525", VAL "72":
      RANDOMIZE USR VAL "15623":
      INK t0:
      CLS
```

- [15] -

